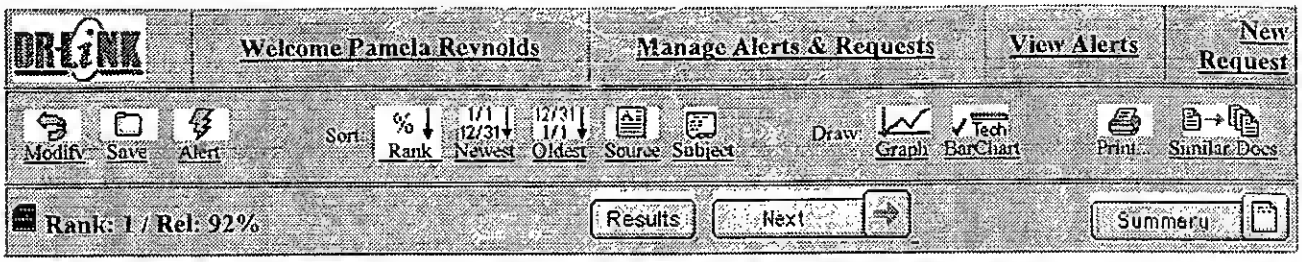

When it comes to choosing a database, the object is value. (object-oriented DBMSs) (includes related article on ObjectWave's DBMS services) (Technology Information)

Watterson, Karen • *Datamation Vol: v44 Issue: n1* • 12/01/97

Most Relevant Section
Document Citation

Object-oriented DBMSs (ODBMS) provide users with several advantages over conventional relational DBMSs (RDBMS), including simpler overall solutions and enhanced support for video, audio and spatial data types. RDBMSs compel users to store data two-dimensionally, whereas ODBMSs allow for the storage of relational data and components. Analysts suggest that ODBMSs are also better suited for handling recursive and telecommunications data and that many human-resource, healthcare, trading and financial systems could benefit from ODBMS technology. ODBMSs accounted for $150 million in overall 1997 database sales, just 3% of the entire market, but analysts project that number to increase dramatically in 1998 with the introduction of innovative new ODBMSs.

Even if you don't have to store data from an atom smasher or track satellites across the sky, a pure object database might help you squeeze more profit fr your data,

PARTS inSIDE OF PARTS. VIDEO IMAGES. Rapidly changing tariff structures. Have you noticed that certain types of data are so complex that it's tough to get full value out of them?

You may not recognize the sound yet, but those data types could be crying out for pure object databases.

Compared to relational databases, object database management systems (ODBMSs) offer much simpler solutions to applications that involve objects and the relationships among them. Now, with native database support for new types of data, such as spatial, audio, and video, and improvements that make it faster to get new ODBMS applications up and running, ODBMSs are becoming even more valuable to the enterprise.

Take Lucent Technologies Customer Technology Support Division as an example: The telephone switch division wanted a global information-sharing system that would improve customer satisfaction and reduce costs by providing better information more quickly to Lucent customer-support people. The existing system was too inflexible to support the growing needs of complex data sharing.

So the division decided to move its customer support and knowledge base off the mainframe into a distributed data-base (replicated to a half-dozen sites worldwide) and to rearchitect the system using an ODBMS from Beaverton, Ore.-based GemStone Systems, rather than a traditional relational database.

According to CEO Sam Cinque-grani of Chicago-based ObjectWave, the software consultant Lucent brought in to build the application, Lucent's goal was to share information on everything related to customers' switches--from configuration and upgrade details to trouble tickets and customer-support planning. GemStone was chosen because of its strength in supporting geographically distributed databases. (See sidebar: "Lucent's choices: the inside story," p. 102.)

The problem with RDBMSs lies in the way they store data: Relational data management systems force you to flatten your data to squeeze it into two-dimensional tables; reassembling it typically requires I/O-intensive joins that also eat CPU cycles.

Lucent isn't the only corporation opting for pure ODBMS technology instead of relational or the newer hybrid object/relational universal databases available from IBM, informix, and Oracle. (See "A database with a head for business," September, p. 62.) Many companies use ODBMSs to store the diverse components (objects) associated with their Web sites-- in much the same way as object-programming pioneers did in the early days of Small-talk and LISP to support complex engineering and manufacturing data in CAD/CAM systems.

The object model: a better fit

Numerous types of data are easier to handle in an object model. Recursive data-- parts within parts- associated with manufacturing is one such data type. Another is telecommunications data containing a navigational element that lets you navigate directly from data element to data element without doing joins to recombine data stored in separate tables. That's why Motorola turned to Objectivity to store the complex celestial information required for its IRIDIUM satellite network.

Objectivity, from the Mountain View, Calif., company of the same name, is used to store what may be the first terabyte ODBMS containing readings from electronic sensors at CERN's collider site in Geneva. According to CERN project leader Jamie Shiers, "We tested the architectural limits of Objectivity/DB, and the only surprises we encountered were good ones." Who says that object databases can't scale?

Healthcare and even human-resource systems also are great candidates for object databases. Groupe Paradis, a retirement-plan management company in France, is using Unidata's recently acquired O2 product to access 100+ gigabytes of data distributed across seven databases.

Cinquegrani, who founded ObjectWave almost a decade ago and has been involved in building dozens of object database systems, notes that finance and trading systems represent another natural application for object databases, because you can build algorithms into the objects. "in the arbitrage market, you need systems that are easy to change in response to new economic conditions," Cinquegrani says.

The rise, fan, and rise of ODBMSs

Object database technology has been around since the late 1970s, when LISP and Smalltalk enjoyed a brief heyday. The heyday was brief because the market wasn't ready. IT departments didn't really understand object-oriented programming languages--never mind object models or object databases-and creating object databases in Smalltalk was

difficult. Adding persistence to object-oriented programming languages seemed like a step backward from mature products like Oracle or DB2--a lot more like programming in COBOL or PL/1 than designing and manipulating relational databases. Worse yet, object databases were viewed as nonscalable, nonmainstream, and hard to query. Some RDBMS aficionados even carped that ODBMSs lacked the mathematical foundations of the relational model.

Object databases made a comeback in the early '90s, when they were touted as the logical successors to RDBMSs. But attitudes change: Two years ago, ODBMSs seemed destilled to bc marginalized out of existence, mere footnotes to the history of databases.

Today, however, a new generation of programmers understands object technology and senses the inevitability of distributed objects. The internet has accelerated the trend toward distributed objects and reignited interest in object databases as RDBMSs evolved.

According to international Data Corp., a research group in Framingham, Mass., the worldwide object database market grew almost 27% in 1996 to more than $100 million. And even with a modest 3% of the estimated 1997 $5 billion database market, the object database market is expected to reach $150 million. With Computer Associates' Jasmine finally hitting the market, those numbers should explode in 1998.

Object Design inc. (ODI) of Burlington, Mass., leads the so-called pure object database market with almost a one-third market share. initially developed with a C++ bias, ODI's ObjectStore is now available with Java, C, and ActiveX interfaces. in a brilliant marketing move, ODI licensed its ObjectStore PSE (persistent storage engine, a native single-user persistent data store) to a number of vendors including Microsoft, Netscape, Novell, and Symantec. That has led a small army of Java developers to learn the ObjectStore API, the same one that's used in ODI's full ObjectStore.

As you might expect from a market leader, ObjectStore has its share of high-profile customers. These include GTE, Southwest Airlines, and Time-Warner, which have built internet applications using ObjectStore. They have discovered that object databases deliver the performance required by dynamic Web applications -applications that typically need to integrate pieces of information on the fly from a variety of sources. ODI markets a handful of utilities, including ObjectForms and inspector, that help programmers build and maintain Web applications.

Another ODI customer is Monterey Bay Aquarium Research institute in Moss Landing, Calif., which develops state-of-the-art equipment and instrumentation and conducts oceanographic research. The institute uses ObjectStore to collect and store multimedia oceanographic research data, including video and image data. The institute replaced its Oracle system with ObjectStore largely because ODI's system made it easy to share research data with other organizations through an extranet application written in Java. ODI continues to exert its leadership position by being one of the first vendors to support Microsoft's OLEDB interface. (Poet, from San Mateo, Calif.-based Poet Software, is a popular low-end ODBMS that has staked out a strong position in the NT environment. Poet supports ActiveX, VB, and integration with Rational's Rose product and plans to ship OLEDB support soon.)

Fremont, Calif.-based Versant is another major player, according to ObjectWave's

Cinquegrani. Vetsant "has the best C++ interface," he says. Cinquegrani also favors GemStone's active server architecture, where objects can execute on the server raffler than having to be shipped and executed on the client, for Java and Smalltalk development. Bin Brook, projects manager for AT&T Labs Europe, agrees. "We are particularly proud of our time to market," he says, describing the bining and access control system his team built using Versant. "Typically, these types of systems take years to develop and require months to change tariff structures." Now, using the Vetsant-based application, Brook's team can modify and ship novel tariff structures within days or weeks.

Did Charles Wang get it right?

In January 1996, when most of the industry was either pooh-poohing object databases or embracing a hybrid object/relational approach, industry analysts scoffed at Computer Associates' Charles Wang when he staked out a position in the ODBMS market. There was a fundamental disconnect between relational databases and object databases, he said. Computer Associates, which recently had acquired ingres, wasn't simply going to extend ingres to serve the needs of object databases; it was going to create a new product.

Two years later, the result of a collaboration with Fujitsu Software is Jasmine.

Computer Associates' entry into the ODBMS market represents a giant vote of confidence in the technology. According to CA senior vice president Yogesh Gupta, "We're in this for the long run." It has an impressive list of more than 30 partners and 50+ beta sites (including L'Oreal, Sony, and Toyota).

Additionally, CA plans to use Jasmine as the preferred object store in its Unicenter TNG enterprise management package. in fact, object databases are nab urals for a repository role, and vendors such as Dulles, Va.-based Template Software are opting for Jasmine as their preferred object store. "Jasmine affords us the ability to map out in-memory object model to a persistent data store," explains Template's Chuck Williams, manager of pre-sales development. "Imagine a complex telecom order application which may take a hundred or more SQL statements to retrieve requested information. With direct mapping to an ODBMS like Jasmine, you only need one call."

Object/relational and universal vendors claim it's easy to store large multimedia objects in relational systems--either directly or via pointers. in fact, most RDBMSs simply store that data in binary large object format, with no special accommodations for querying it.

The differences between RDBMSs and ODBMSs are more than skin deep. Both support transactions (all-or-nothing commits), and constructs such as relationships, cursors, and indexes. Object databases, thanks to inheritance, are intrinsically extensible and support a wide array of multivalued structures like lists, arrays, and bags (a bag being an unordered aggregation of objects in which there may be duplicates). Anyone who has tried to model bin-of-materials-type applications within the RDBMS model will appreciate object databases' navigational data access.

Many studies and surveys show that customers are using ODBMSs to deploy new applications, and they're doing it because of reduced time to market. Most ODBMS vendors view Oracle as a fierce and intimidating competitor, but the failure of Sedona (the object development environment Oracle was developing but abandoned) has helped

the pure ODBMS industry. Firms cite the fundamental mismatch between object and relational technology, along with easy schema evolution and performance, as reasons to opt for pure ODBMSs.

Standards and benchmarks

SQL is virtually synonymous with RDBMSs, and most ODBMSs also support SQL, but the ODBMS market has its own set of standards, notably the object database standard, ODMG 2.0, and OQL. The Object Database Management Group's second-generation standard, ODMG 2.0, builds upon the Object Management Group's fundamental CORBA-compliant object model and is the basis for almost all vendors' C++, Smalltalk, and Java bindings. ODMG 2.0 also defines OQL as a superset of SQL92, and OQL is finally beginning to enjoy industry support.

Although the object benchmark scene isn't as mature as the ODMG and OQL standards, there are more than a dozen object benchmarks, including OO1, HyperModel, and007. They're not as well known as their RDBMS counterparts from the Transaction Processing Council, but most potential ODBMS adopters aren't worried about benchmarking anyway. Whether they face unacceptably poor performance with their relational systems or simply realize how hard it will be to map their new object-oriented apps to the relational model, ODBMS adopters realize how difficult it is to develop meaningfull benchmarks, especially in trying to compare relational or object/relational to pure object solutions.

Karen Watterson is a San Diego-based writer and consultant specializing in database and data warehousing issues. She writes a column for Windows NT Magazine, is editor of Pinnacle Publishing's "Visual Basic Developer" and "SQL Server Professional" newsletters, and has written two books for Addison-Wesley: Visual Basic Database Programming and Client/Server Technology for Managers. Her address is karen--watterson@msn.com.

Additional Information:

Audience: General; Trade
Publication Format: Magazine/Journal
Industry Category: BUSN Any type of business; CMPT Computers and Office Automation
Identifier(s): Object-Oriented Database; Technology Overview; Technology Application
Illustration: table
ISSN: 0011-6963
Product Code/Name: 7372421 (DBMS)
Record Date: 980302
Record Number: 20165219
SIC Description: 7372 Prepackaged software
Subject Heading: Object-oriented data bases--Design and construction; Data base industry--Marketing

Word Count: 2029

Document Rank: 1

Headline/Title: When it comes to choosing a database, the object is value. (object-oriented DBMSs) (includes related article on ObjectWave's DBMS services) (Technology Information)

Author(s): Watterson, Karen

Date: 12/01/97

Source: Datamation

Volume/Issue/Pg: Vol: v44 Issue: n1

Database: IAC Consolidated Business Collection

Num. Pages: 9 (2400 words)

Accession Num: 20165219

MNIS Document: 55029

When it comes to choosing a database, the object is value. (object-oriented DBMSs) (includes related article on ObjectWave's DBMS services) (Technology Information)

Watterson, Karen • *Datamation Vol: v44 Issue: n1* • 12/01/97

Most Relevant Section
Document Citation

Object-oriented DBMSs (ODBMS) provide users with several advantages over conventional relational DBMSs (RDBMS), including simpler overall solutions and enhanced support for video, audio and spatial data types. RDBMSs compel users to store data two-dimensionally, whereas ODBMSs allow for the storage of relational data and components. Analysts suggest that ODBMSs are also better suited for handling recursive and telecommunications data and that many human-resource, healthcare, trading and financial systems could benefit from ODBMS technology. ODBMSs accounted for $150 million in overall 1997 database sales, just 3% of the entire market, but analysts project that number to increase dramatically in 1998 with the introduction of innovative new ODBMSs.

Even if you don't have to store data from an atom smasher or track satellites across the sky, a pure object database might help you squeeze more profit fr your data,

PARTS inSIDE OF PARTS. VIDEO IMAGES. Rapidly changing tariff structures. Have you noticed that certain types of data are so complex that it's tough to get full value out of them?

You may not recognize the sound yet, but those data types could be crying out for pure object databases.

Compared to relational databases, object database management systems (ODBMSs) offer much simpler solutions to applications that involve objects and the relationships among them. Now, with native database support for new types of data, such as spatial, audio, and video, and improvements that make it faster to get new ODBMS applications up and running, ODBMSs are becoming even more valuable to the enterprise.

Take Lucent Technologies Customer Technology Support Division as an example: The telephone switch division wanted a global information-sharing system that would improve customer satisfaction and reduce costs by providing better information more quickly to Lucent customer-support people. The existing system was too inflexible to support the growing needs of complex data sharing.

So the division decided to move its customer support and knowledge base off the mainframe into a distributed data-base (replicated to a half-dozen sites worldwide) and to rearchitect the system using an ODBMS from Beaverton, Ore.-based GemStone Systems, rather than a traditional relational database.

According to CEO Sam Cinque-grani of Chicago-based ObjectWave, the software consultant Lucent brought in to build the application. Lucent's goal was to share information on everything related to customers' switches--from configuration and upgrade details to trouble tickets and customer-support planning. GemStone was chosen because of its strength in supporting geographically distributed databases. (See sidebar: "Lucent's choices: the inside story," p. 102.)

The problem with RDBMSs lies in the way they store data: Relational data management systems force you to flatten your data to squeeze it into two-dimensional tables; reassembling it typically requires I/O-intensive joins that also eat CPU cycles.

Lucent isn't the only corporation opting for pure ODBMS technology instead of relational or the newer hybrid object/relational universal databases available from IBM, informix, and Oracle. (See "A database with a head for business," September, p. 62.) Many companies use ODBMSs to store the diverse components (objects) associated with their Web sites-- in much the same way as object-programming pioneers did in the early days of Small-talk and LISP to support complex engineering and manufacturing data in CAD/CAM systems.

The object model: a better fit

Numerous types of data are easier to handle in an object model. Recursive data-- parts within parts- associated with manufacturing is one such data type. Another is telecommunications data containing a navigational element that lets you navigate directly from data element to data element without doing joins to recombine data stored in separate tables. That's why Motorola turned to Objectivity to store the complex celestial information required for its IRIDIUM satellite network.

Objectivity, from the Mountain View, Calif., company of the same name, is used to store what may be the first terabyte ODBMS containing readings from electronic sensors at CERN's collider site in Geneva. According to CERN project leader Jamie Shiers, "We tested the architectural limits of Objectivity/DB, and the only surprises we encountered were good ones." Who says that object databases can't scale?

Healthcare and even human-resource systems also are great candidates for object databases. Groupe Paradis, a retirement-plan management company in France, is using Unidata's recently acquired O2 product to access 100+ gigabytes of data distributed across seven databases.

Cinquegrani, who founded ObjectWave almost a decade ago and has been involved in building dozens of object database systems, notes that finance and trading systems represent another natural application for object databases, because you can build algorithms into the objects. "in the arbitrage market, you need systems that are easy to change in response to new economic conditions," Cinquegrani says.

The rise, fan, and rise of ODBMSs

Object database technology has been around since the late 1970s, when LISP and Smalltalk enjoyed a brief heyday. The heyday was brief because the market wasn't ready. IT departments didn't really understand object-oriented programming languages--never mind object models or object databases-and creating object databases in Smalltalk was

difficult. Adding persistence to object-oriented programming languages seemed like a step backward from mature products like Oracle or DB2--a lot more like programming in COBOL or PL/1 than designing and manipulating relational databases. Worse yet, object databases were viewed as nonscalable, nonmainstream, and hard to query. Some RDBMS aficionados even carped that ODBMSs lacked the mathematical foundations of the relational model.

Object databases made a comeback in the early '90s, when they were touted as the logical successors to RDBMSs. But attitudes change: Two years ago, ODBMSs seemed destilled to be marginalized out of existence, mere footnotes to the history of databases.

Today, however, a new generation of programmers understands object technology and senses the inevitability of distributed objects. The internet has accelerated the trend toward distributed objects and reignited interest in object databases as RDBMSs evolved.

According to international Data Corp., a research group in Framingham, Mass., the worldwide object database market grew almost 27% in 1996 to more than $100 million. And even with a modest 3% of the estimated 1997 $5 billion database market, the object database market is expected to reach $150 million. With Computer Associates' Jasmine finally hitting the market, those numbers should explode in 1998.

Object Design inc. (ODI) of Burlington, Mass., leads the so-called pure object database market with almost a one-third market share. initially developed with a C++ bias, ODI's ObjectStore is now available with Java, C, and ActiveX interfaces. in a brilliant marketing move, ODI licensed its ObjectStore PSE (persistent storage engine, a native single-user persistent data store) to a number of vendors including Microsoft, Netscape, Novell, and Symantec. That has led a small army of Java developers to learn the ObjectStore API, the same one that's used in ODI's full ObjectStore.

As you might expect from a market leader, ObjectStore has its share of high-profile customers. These include GTE, Southwest Airlines, and Time-Warner, which have built internet applications using ObjectStore. They have discovered that object databases deliver the performance required by dynamic Web applications -applications that typically need to integrate pieces of information on the fly from a variety of sources. ODI markets a handful of utilities, including ObjectForms and inspector, that help programmers build and maintain Web applications.

Another ODI customer is Monterey Bay Aquarium Research institute in Moss Landing, Calif., which develops state-of-the-art equipment and instrumentation and conducts oceanographic research. The institute uses ObjectStore to collect and store multimedia oceanographic research data, including video and image data. The institute replaced its Oracle system with ObjectStore largely because ODI's system made it easy to share research data with other organizations through an extranet application written in Java. ODI continues to exert its leadership position by being one of the first vendors to support Microsoft's OLEDB interface. (Poet, from San Mateo, Calif.-based Poet Software, is a popular low-end ODBMS that has staked out a strong position in the NT environment. Poet supports ActiveX, VB, and integration with Rational's Rose product and plans to ship OLEDB support soon.)

Fremont, Calif.-based Versant is another major player, according to ObjectWave's

Cinquegrani. Vetsant "has the best C++ interface," he says. Cinquegrani also favors GemStone's active server architecture, where objects can execute on the server raffler than having to be shipped and executed on the client. for Java and Smalltalk development. Bin Brook, projects manager for AT&T Labs Europe, agrees. "We are particularly proud of our time to market," he says, describing the bining and acccss control system his team built using Versant. "Typically, these types of systems take years to develop and require months to change tariff structures." Now, using the Vetsant-based application, Brook's team can modify and ship novel tariff structures within days or weeks.

Did Charles Wang get it right?

In January 1996, when most of the industry was either pooh-poohing object databases or embracing a hybrid object/relational approach, industry analysts scoffed at Computer Associates' Charles Wang when he staked out a position in the ODBMS market. There was a fundamental disconnect between relational databases and object databases, he said. Computer Associates, which recently had acquired ingres, wasn't simply going to extend ingres to serve the needs of object databases; it was going to create a new product.

Two years later, the result of a collaboration with Fujitsu Software is Jasmine.

Computer Associates' entry into the ODBMS market represents a giant vote of confidence in the technology. According to CA senior vice president Yogesh Gupta, "We're in this for the long run." It has an impressive list of more than 30 partners and 50+ beta sites (including L'Oreal, Sony, and Toyota).

Additionally, CA plans to use Jasmine as the preferred object store in its Unicenter TNG enterprise management package. in fact, object databases are nab urals for a repository role, and vendors such as Dulles, Va.-based Template Software are opting for Jasmine as their preferred object store. "Jasmine affords us the ability to map out in-memory object model to a persistent data store," explains Template's Chuck Williams, manager of pre-sales development. "Imagine a complex telecom order application which may take a hundred or more SQL statements to retrieve requested information. With direct mapping to an ODBMS like Jasmine, you only need one call."

Object/relational and universal vendors claim it's easy to store large multimedia objects in relational systems--either directly or via pointers. in fact, most RDBMSs simply store that data in binary large object format, with no special accommodations for querying it.

The differences between RDBMSs and ODBMSs are more than skin deep. Both support transactions (all-or-nothing commits), and constructs such as relationships, cursors, and indexes. Object databases, thanks to inheritance, are intrinsically extensible and support a wide array of multivalued structures like lists, arrays, and bags (a bag being an unordered aggregation of objects in which there may be duplicates). Anyone who has tried to model bin-of-materials-type applications within the RDBMS model will appreciate object databases' navigational data access.

Many studies and surveys show that customers are using ODBMSs to deploy new applications, and they're doing it because of reduced time to market. Most ODBMS vendors view Oracle as a fierce and intimidating competitor, but the failure of Sedona (the object development environment Oracle was developing but abandoned) has helped

the pure ODBMS industry. Firms cite the fundamental mismatch between object and relational technology, along with easy schema evolution and performance, as reasons to opt for pure ODBMSs.

Standards and benchmarks

SQL is virtually synonymous with RDBMSs, and most ODBMSs also support SQL, but the ODBMS market has its own set of standards, notably the object database standard, ODMG 2.0, and OQL. The Object Database Management Group's second-generation standard, ODMG 2.0, builds upon the Object Management Group's fundamental CORBA-compliant object model and is the basis for almost all vendors' C++, Smalltalk, and Java bindings. ODMG 2.0 also defines OQL as a superset of SQL92, and OQL is finally beginning to enjoy industry support.

Although the object benchmark scene isn't as mature as the ODMG and OQL standards, there are more than a dozen object benchmarks, including OO1, HyperModel, and007. They're not as well known as their RDBMS counterparts from the Transaction Processing Council, but most potential ODBMS adopters aren't worried about benchmarking anyway. Whether they face unacceptably poor performance with their relational systems or simply realize how hard it will be to map their new object-oriented apps to the relational model, ODBMS adopters realize how difficult it is to develop meaningfull benchmarks, especially in trying to compare relational or object/relational to pure object solutions.

Karen Watterson is a San Diego-based writer and consultant specializing in database and data warehousing issues. She writes a column for Windows NT Magazine, is editor of Pinnacle Publishing's "Visual Basic Developer" and "SQL Server Professional" newsletters, and has written two books for Addison-Wesley: Visual Basic Database Programming and Client/Server Technology for Managers. Her address is karen--watterson@msn.com.

Additional Information:

Audience: General; Trade
Publication Format: Magazine/Journal
Industry Category: BUSN Any type of business; CMPT Computers and Office Automation
Identifier(s): Object-Oriented Database; Technology Overview; Technology Application
Illustration: table
ISSN: 0011-6963
Product Code/Name: 7372421 (DBMS)
Record Date: 980302
Record Number: 20165219
SIC Description: 7372 Prepackaged software
Subject Heading: Object-oriented data bases--Design and construction; Data base industry--Marketing

Word Count: 2029

Document Rank: 1

Headline/Title: When it comes to choosing a database, the object is value. (object-oriented DBMSs) (includes related article on ObjectWave's DBMS services) (Technology Information)

Author(s): Watterson, Karen

Date: 12/01/97

Source: Datamation

Volume/Issue/Pg: Vol: v44 Issue: n1

Database: IAC Consolidated Business Collection

Num. Pages: 9 (2400 words)

Accession Num: 20165219

MNIS Document: 55029

ActiveX vs. Java. (includes related article on the programming languages' security concerns) (Industry Trend or Event)(Cover Story)

Stirland, Sarah • *Wall Street & Technology Vol: v15 Issue: n8* • 08/01/97

The Java and ActiveX programming languages offer a range of individual benefits that are attracting support from different factions within the computing industry, but the technologies are best used when deployed in specific environments. Financial systems integrators supportJava's technological advantages in cross-platform computing, but many suggest that the technology is presently too immature and inadequately understood to be fully useful. The technology behind ActiveX is more fully understood by the majority of working developers, but the fact that ActiveX is designed for use in environments running Microsoft's Windows platform alienates some systems integrators. Companies already running Unix and associated technologies are likely to support Java, whereas firms that are committed to Windows are more likely to opt for ActiveX.

Financial institutions that are flocking to use Sun Microsystem's Java tout its portability but criticize its immaturity and while there are more developers with knowledge of ActiveX tools, its exclusivity to Microsoft is a problem.

As the national debate on the relative benefits of two mobile computing languages, Java and ActiveX, swings back and forth in newspapers,magazines and Internet bulletin boards, financial systems technologists are quietly conducting their own tests and coming to their own conclusions.

The consensus, among those who have experimented with the two systems, is that the debate isn't simply about one system usurping the roleof another.

Sun Microsystem's Java language versus Microsoft Corp.'s ActiveX system is about the exploration of tantalizing new opportunities versus the reliance on more mature, and thus perceived as more stable, technology.

ActiveX and Java are two different types of portable code that are being used to integrate live content, graphics and animation into applications for deployment in network computing.

The majority of users who talked to Wall Street & Technology and whohave experimented with the two languages say that it's a mistake to portray one as "better" than another, often for the reason that the languages usually serve their purposes in a variety of different situations. In general, firms say Java is a good way to reach out to places they had a hard -- and expensive time reaching before. Conversely, ActiveX is

often used for applications running in existing Windows environments.

For example, two leaders who are spearheading the Java development efforts at BankBoston are pinning their hopes on Java's ability to runacross worldwide intranets and extranets.

The developers, whose efforts mirror many others on Wall Street, want to use Java to help the firm save money at the same time it beefs up the capabilities of their global operations.

"We have a group of about 15 to 20 Unix system administrators supporting this environment, not to mention the equipment, the consulting and the market data and everything else," says Mike Siegenfeld, director of the derivatives and emerging markets systems at BankBoston's global capital markets group.

"We're looking for a way to do that more cheaply so that we can haveall our resources in one place here," he says.

Siegenfeld and Pam Haller, the head developer for the emerging markets applications, are working on rolling out an emerging markets Java applet called Emgraph in all of the group's offices in Singapore, Mexico City and Seoul, South Korea by the end of the year. Almost 20 traders in both Boston and London already use the system, which was up and running on those trading floors this past April.

The Java applets should allow both the traders in these remote offices and clients to access financial engineering applications written in The Mathworks, Inc.'s Matlab software through an extranet or an intranet.

The plan is to provide global access to BankBoston's proprietary historical analytics through applets on the intranet. The system will use the Unix-based messaging middleware Corba to link up to servers in Boston, where the hardcore number crunching is done with the more powerful C++ code.

Following the successful deployment of the infrastructure, Haller says BankBoston plans on upgrading the system to handle real-time analytics.

"This is an example of how we're using Java and the Internet to basically gain a global application capability without having a global infrastructure -- it's a thin client design, which is an overused word,but it works," Siegenfeld comments.

One important problem that Java should solve for BankBoston is that of deploying such complex programs across a smorgasbord of operating systems across the world, says Siegenfeld. BankBoston uses a variety of Unix-based Sun workstations at its headquarters but its overseas offices rely on PCs.

Once this analytics system is up and running, Siegenfeld and Haller plan on implementing derivatives and emerging market bond trading capabilities on the same system, either by the end of the year or by thefirst quarter of next year. Ultimately, the two hope that this system will allow the firm to route all the back office duties resulting from these transactions to their Boston headquarters.

"We're hoping that the entire bank can use a single back office based in Boston to support our entire organization," Siegenfeld says.

Although Siegenfeld says that he's had "heated discussions," within his firm about Java and ActiveX, the latter was never even consideredan option because of BankBoston's reliance on Unix.

But Siegenfeld says he never would have considered ActiveX anyway because the whole point of being able to reach out without having to worry about the end-users' operating system and local support would have been lost.

Most other securities firms like Java for the same reason of portability, say financial technology consultants. Other object-oriented languages, namely C++, which have similar properties, are more powerful,more mature and well documented but don't feature this crucial element. In addition, ActiveX components are more widely used for Windows applications, they say.

"ActiveX is more oriented toward gluing desktops together so that I can write applications that pump things off my desktop onto yours," says Dr. Andrew Herbert, head of a European research team on distributed systems technology and a chief technology officer at Digitivity, Inc., a network management firm in Los Altos, Calif. and Cambridge, England.

But Java is more often used in conjunction with Corba to link servers, Herbert notes.

"What you find people doing is gluing their desktops together with ActiveX, connecting from their desktops to their back office servers using Corba and then gluing their servers together with Corba [as well,] so they're using a hybrid approach," he says.

The consulting firm Fusion Systems Group, Inc., which operates a Sundesignated Object Reality Center and consults with BankBoston, has actually helped financial firms integrate such Java-related and ActiveX systems, says Steve Siegal, the firm's managing director. The pointis not to anchor the systems to ActiveX.

"It's important to understand that the Java/Corba model includes ActiveX as a technology that they're happy to work with -- but ActiveX is exclusionary, so if you base your architecture on ActiveX, you limit your choices," says Siegal.

Michael Synnott, an equities and futures trading support technologist in Merrill Lynch & Co.'s London office, agrees: "If you do use ActiveX, you are bolted into Microsoft -- end of story."

Synnott was recently part of a team that had to decide whether to use Java or ActiveX for the foundation of an internal application. The team ended up choosing Java.

The decision was based on the amount of support that the ActiveX components would have required compared to the Java applets, Synnott says. After performing some stress-testing, the team found that Java required fewer lines of source code than ActiveX.

In addition, the ActiveX solution required Merrill to install class libraries on every end-user's workstation, unlike Java applets, whichonly had to be installed on a central server.

But Synnot intends to continue to use ActiveX controls to write the graphical front-ends for some of the trading desk's financial analytics systems. He finds that ActiveX can often be a more palatable solution because more tools are available. Others will for now choose thisroute for the same reasons, he predicts.

"At the moment, a lot of people will shy away from Java because you've got very few people who've got any great knowledge of it and who are going to support it afterwards," he says, but BankBoston's Siegenfeld countered this problem by hiring C++ programmers who he then has trained in Java. Seven programmers have been trained through intensive workshops that can take up to a month, he says. Six more remain to be trained.

The number of tools available and their reliability were certainly concerns for the developers of DLJ Direct, formerly known as PC Financial Network, says Suresh Kumar, DLJ Direct's chief technology officer.

Where the features of Java seemed to offer BankBoston solutions, it seems to do the opposite for DLJ Direct's on-line trading system.

The on-line brokerage, which has more than 352,000 accounts, went the other direction from the open systems on the Web this May when it gave its frequent traders the additional option of trading through a test version of its proprietary Windows-based front-end system. The firm bought ActiveX tools for the system's animation and charting features.

Kumar says that HTML was too cumbersome and Java is still too immature.

"There are charting tools available for Java, but Windows has been around longer than Java," Kumar says. "The charting package that we use is version five but with Java, it's likely to be version one."

Another consideration was Java's limited functionality.

With ActiveX, "we can take advantage of the caching information as well as going to our own middleware to make the transactions faster --for example, if somebody got a chart at 4:15 and they go back and ask for the same chart at five o'clock, why should they go back to the server? In Java, since you have no control over access to the local machine, you can't do that," he says.

"The other thing is that every time you access Java, it has to download the code as well as the data, whereas with ActiveX you download it once and you don't have to get the code again."

For Kumar; the more mature off-the-shelf ActiveX tools were a cheaper and more reliable alternative to both the problems of immature Javatools and a team of expensive Java programmers.

Sun says it is working on several new initiatives that will improve Java's widely publicized performance problems. For now, however, developers, including BankBoston's Siegenfeld, admit that performance is still an issue, which is why firms typically run Java clients with C++ servers backing them up.

"There aren't a lot of huge applets that people are using on a day to day basis," currently due to a number of performance-based problems, notes David Osborne, managing director of a New York-based consulting firm Micro Modeling Associates.

Network Design Should be Part of Security Solution, Say Experts

By now, the stories of the security hazards associated with Java applets and ActiveX components are legion.

And as financial firms launch into doing business through these new systems, they must also address the widely reported security issues, which are most commonly buggy applets that might damage an end-user'shard drive.

For a vivid demonstration of what Microsoft Corp.'s ActiveX is capable of, users could go to a site called the "Exploder," at http://www.halcyon.com/mclain/ActiveX/. The site, which is a demonstration designed by a computer consultant, contains an ActiveX component that if downloaded shuts down Windows 95 users' computers.

Another site to check out is Sun Microsystems' subsidiary Javasoft'ssecurity pages at: http://java.sun.com/sfaq/which catalogues a long list of security bugs and fixes associated with the Java Development Kit 1.1.

Experts say that firms that plan on including mobile code such as Java and ActiveX in their forays into the world of electronic commerce should have their security policy built into their network -- rather than sticking it on afterwards like a bandage.

Digital signatures, firewalls, and other individual approaches to the issue of network security are piecemeal attempts to solve a multi-faceted problem, they say.

"People should think of the overall security architecture of their system," says Marianne Mueller, an engineer at Javasoft, "because it'soften the interaction of components that can lead to interesting loopholes."

That means that firewall solutions should only make up one part of alarger security policy.

Firewall vendors' have typically offered two types of solutions. Oneis to completely block mobile code, such as Java and ActiveX, from entering a firm's internal network and the other is to apply a virus-like scan to incoming code that check for patterns that might reveal amalicious application.

"I haven't seen products that 'solve' the problem," Mueller says.

One new product introduced this June called "The Cage" aims to tackle these problems by adopting the network architecture approach.

The Cage is a network architecture system that routes all an enterprise's incoming applets to a server with a Java virtual machine. The server acts as a "cage" that isolates the mobile code. End-users access the actual applet on the server through a "proxy applet," which is installed on their desktop. Their hard drives are therefore protectedfrom any malicious or faulty code, according to Dr. Andrew Herbert, chief technology officer at the network management firm Digitivity, Inc., which has offices both in Los Altos, Calif. and Cambridge, England.

The system grew out of a project based at the London offices of the investment bank SBC Warburg that was specifically designed to deal with applets, although it could be used for other types of code, says Herbert.

SBC Warburg was pressed to find a solution because it had been receiving angry responses from its trading partners, Herbert says.

"Stockbrokers discovered that Swiss Bank had a policy of not lettingJava cross its firewall," he says. "They therefore said to Swiss Bank "Why should I run your Java when you won't run mine?"

SBC Warburg is an investment banking subsidiary of Swiss Bank. The investment bank has started a pilot project that replaces its internalexisting front-end client applications for trading with Java, says Herbert, and has also started to use Java to trade with its clients.

Additional Information:

Audience: Trade
Publication Format: Magazine/Journal
Industry Category: BANK Banking, Finance and Accounting; BUSN Any type of business; CMPT Computers and Office Automation
Identifier(s): Java; ActiveX; Market Trend/Market Analysis; Programming Language; Industry Trend
Illustration: photograph
Article Type: Cover Story
Record Date: 970822
Record Number: 19685017
Word Count: 2233

Document Rank: 5

Headline/Title: ActiveX vs. Java. (includes related article on the programming languages' security concerns) (Industry Trend or Event)(Cover Story)

Author(s): Stirland, Sarah

Date: 08/01/97

Source: Wall Street & Technology

Volume/Issue/Pg: Vol: v15 Issue: n8

Database: IAC Computer Collection

Num. Pages: 9 (2600 words)

Accession Num: 19685017

MNIS Document: 45881